Spherly User Manual

Spherly is a visual programming editor application built upon **Blockly**, which supports the creation of new block languages (https://code.google.com/p/blockly/). Spherly was designed to allow programmatic command of the Sphero Robot Ball (http://www.gosphero.com/) through intuitive visual "block" commands that can be arranged sequentially in a program.

Spherly has two components: the front-end web application that runs from a browser, and the local server that runs on your computer and communicates with Sphero via Bluetooth. The web application and the server are both required, and can be found at: http://outreach.cs.ua.edu/spherly/**.**

I. Starting the Server

From the website listed above, clicking the "Download Server" link will download a zip file containing the necessary files to run the server application on your computer. After unzipping the file, simply run the SpherlyServer.exe application inside of it, and the server should begin to execute. The server needs to be running on your computer whenever you want to use Spherly.

The server currently only works on Windows machines with Bluetooth communication (tested on Windows 7 and Windows 8), but we are working on installations for tablets and other operating system. The front-end web interface will work with any computer, but the back-end local server needs to be customized for each platform.

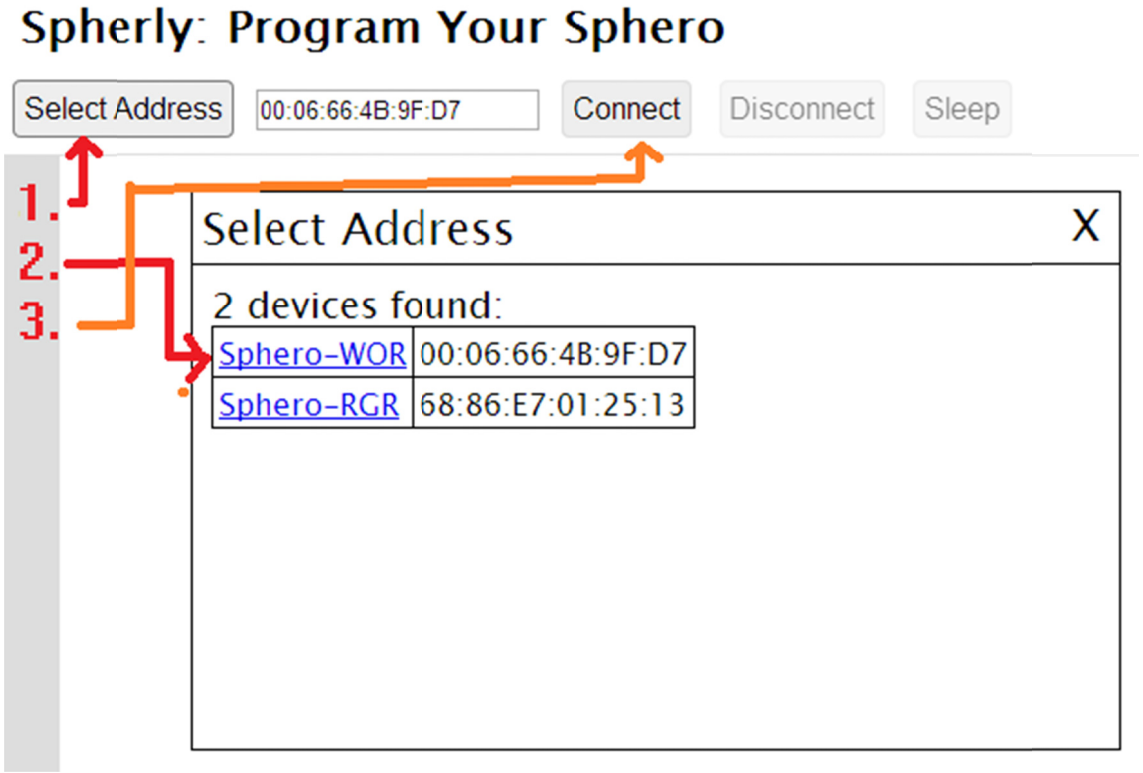
II. Connecting to Sphero

After starting the local server, you must connect to the Sphero through the Windows Bluetooth manager.

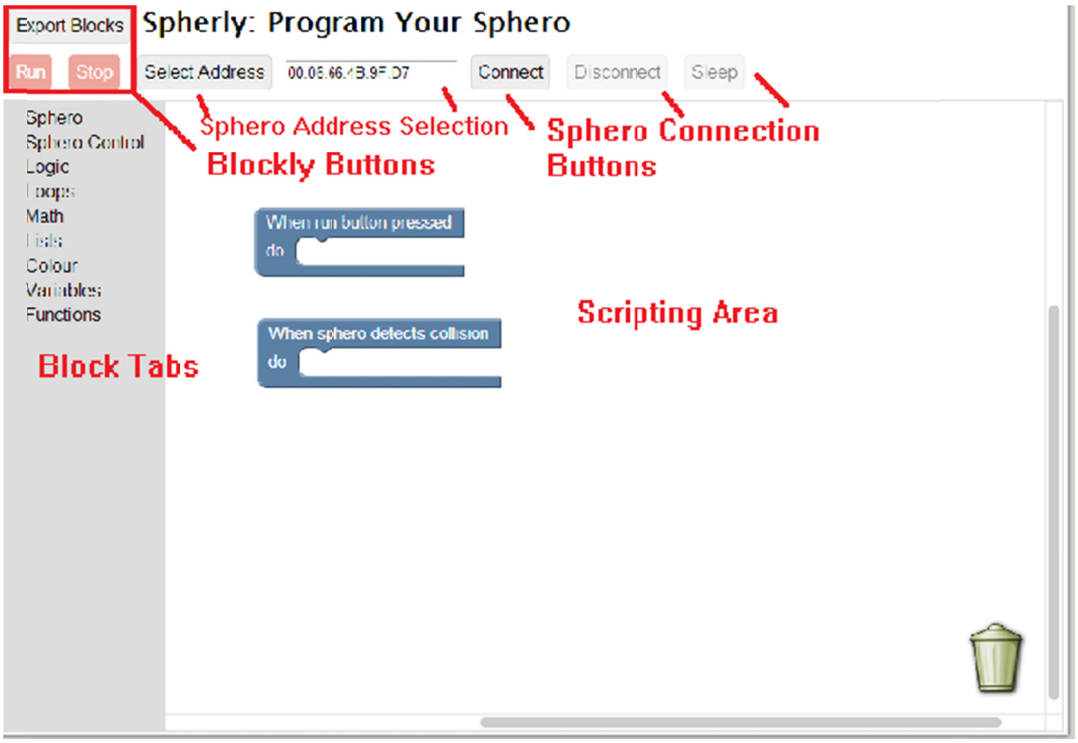
After your Server and Bluetooth are set up, navigate to the Spherly Application (available from the main Spherly web page, or directly at http://outreach.cs.ua.edu/spherly/run/) and perform the following steps:

1. Click the Select Address Button
2. Select the appropriate Sphero you want to connect to through the popup window
3. Click Connect

You may have to try connecting more than once if the Server just started. Sphero can only be connected to one computer or application at a time, but the app should notify you when you're connected to a Sphero successfully. A connected Sphero should light up with a blue color.



III. Layout of Spherly Web Application



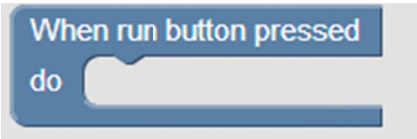
After connecting to the Sphero, the majority of the work will be done with the **Blockly Buttons**, **Scripting Area**, and the **Block Tabs**. You can drag command Blocks over from the *Block Tabs* to the *Scripting Area* into a sequence of commands to be executed by Sphero. All blocks that you want Sphero to execute must be contained either in the "**When run button pressed**" container block or the "**When sphero detects collision**" container block.

When you have your blocks set up in the program you want, simply press the **Run** button to send the instructions from the web client to the server, and from there to the Sphero ball.

IV. Spherly Specific Blocks

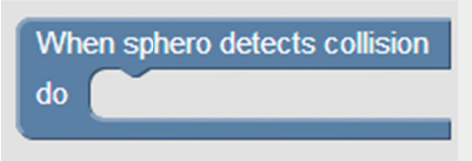
Now we'll go through all of the blocks available for creating your Spherly application.

A. Blocks under the "Sphero Control" Tab



This is one of the two blocks that are present in the Scripting Area when Spherly starts. This container block will hold all blocks that you want to execute on the Sphero when the **Run Button** is pressed.

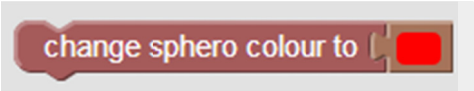
Any blocks that are outside of the "When run button" block will not have any direct effect on the Sphero. (**NOTE:** an exception is any **function** created outside of the *run* block. These will not be executed directly, but can be referenced from inside of the *run* block and will function properly).



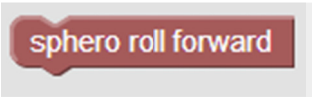
This is the other block that is present in the Scripting Area when Spherly starts. Similarly to the "When run button" block, this container block will hold all blocks that you want to command the Sphero when it detects a collision (e.g., running into a wall or being lightly hit).

Collision detection in this version of Spherly is somewhat sensitive, and so it is common for Sphero to detect a collision when it begins rolling or turning on its own. This is a lot more present on certain floor surfaces than others. If you don't put any blocks in this wrapping block, or delete this wrapping block, Sphero will not do anything when it collides with something.

B. Blocks under the "Sphero" Tab

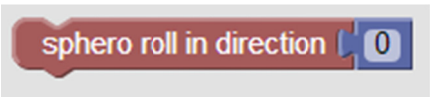


This block will make Sphero's LED light up to the specified color. The color portion on the right can be changed using the built in color picker (click on the color rectangle to bring up the picker), or switched out with any block from the **Colour** tab.



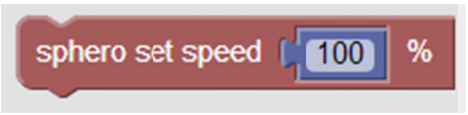
This block will instruct Sphero to begin rolling forward. Sphero calibrates its positioning upon starting, so it may be helpful to use this command when you first start the application to find out what direction Sphero thinks forward is.

(**Note:** the direction Sphero thinks is "forward" will be changed by using the "**turn**" command below)

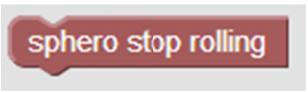


This block will instruct Sphero to roll in the specified direction. The direction is in units of degrees, and corresponds to the classic 0 to 359 degrees of a circle. Rolling in the direction of 0 degrees corresponds to "roll forward" if you have not yet used the "**turn**" command.

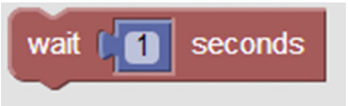
This block instructs Sphero to change its **forward** direction by the set amount. If Sphero is instructed to turn 180 degrees, it will be facing the opposite direction it was previously facing, and any further "**roll forward**" commands will use this new facing.



Changes the speed of Sphero to a percentage of its max speed when executed. Sphero's initial speed is set at 100%. Max speed varies depending on which Spher version you are using (Sphero 1.0 and Sphero 2.0).



If Sphero is currently rolling, this will make it stop at its current location. If Sphero is not rolling, this command does nothing.

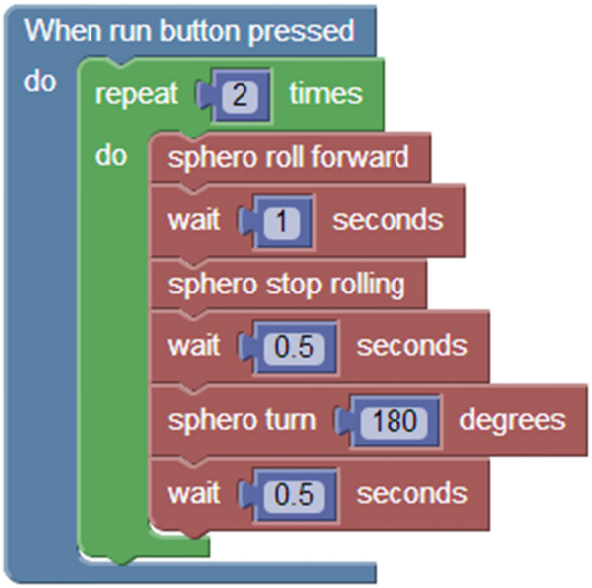


While this will not send a specific command to Sphero, it can be used to pause before you send the next command to Sphero. For instance, because Sphero takes half a second to turn, it could be helpful to put a "**wait**" command after a "**turn**" command, in order to give the Sphero enough time to position itself correctly. If this is not done, Sphero may not roll in the correct direction immediately.

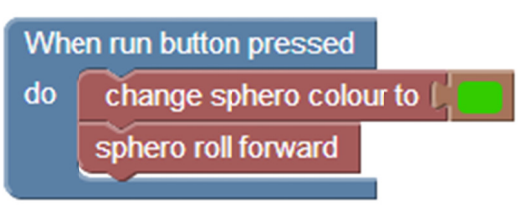
V. Code Examples

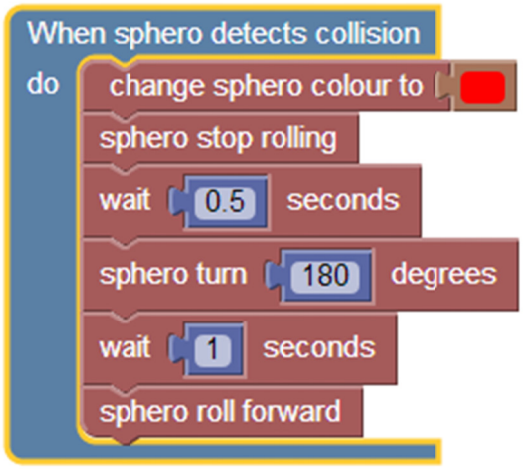
Note: When working with Sphero, the robot can sometimes become confused when trying to "right itself upward." Therefore, it is not guaranteed that every execution of the code will run the same every time. Allowing Sphero more time to execute *turn* commands by using *wait* commands is suggested, but may have variations of accuracy.

A. About Face

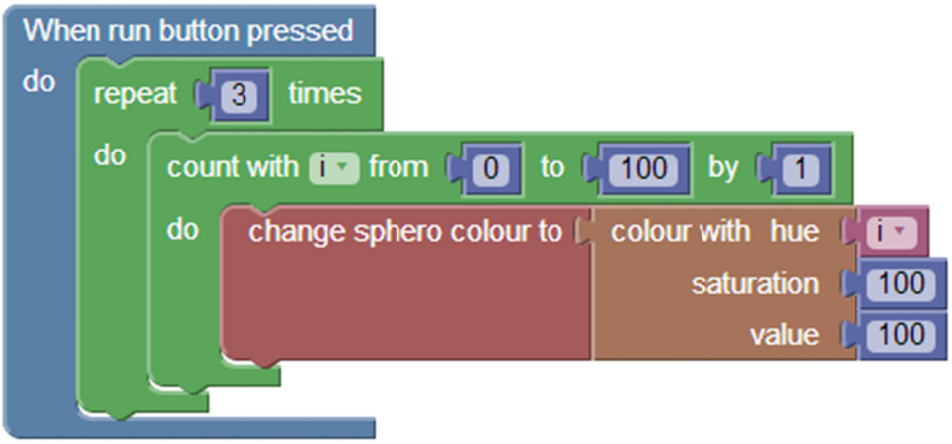


B. About Face: Collision Edition

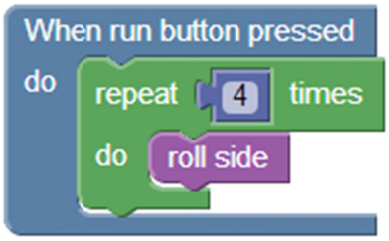


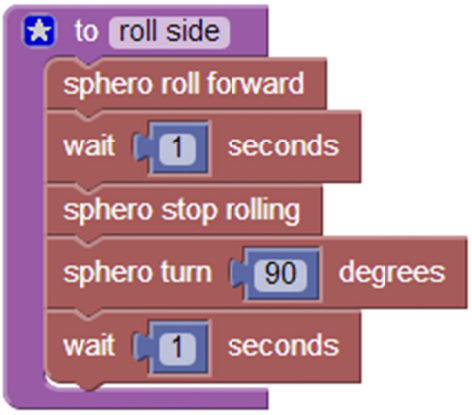


C. Color Cycling

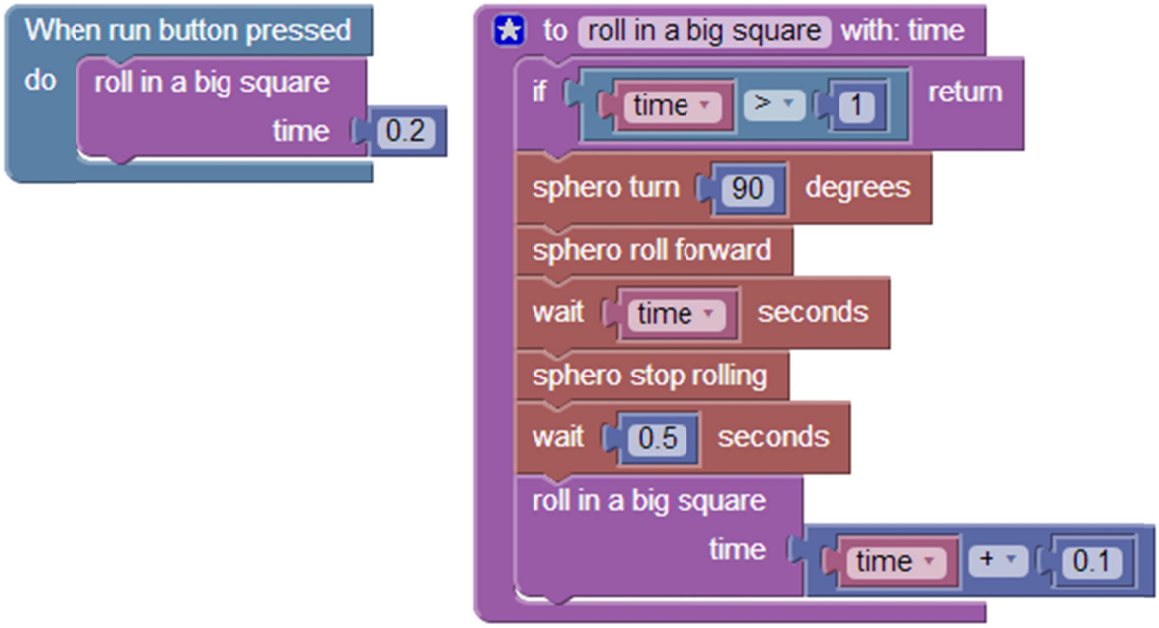


D. Rolling in a Square





E. Rolling in a Spiraling Square



VI. Spherly User's Forum

To ask questions or post comments, please visit the Spherly user forum at:

http://tinyurl.com/spherly-forum